Rule WLM350: I/O activity may have caused significant delays

Finding: CPExpert believes that I/O activity by the service class may be a significant cause of the service class missing its performance goal.

This finding applies only to MVS versions prior to OS/390 Release 3. I/O activity and I/O delays were added to SMF Type 72 records with OS/390 Release 3. Prior to OS/390 Release 3, I/O activity was included in the UNKNOWN category of delay. WLM350(series) rules were designed to estimate I/O problems when a service class had a significant amount of UNKNOWN delay.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT, depending upon the amount of I/O activity and the delay to the service class caused by the I/O activity.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101: Service Class did not achieve average response goal
Rule WLM102: Service Class did not achieve percentile response goal
Rule WLM103: Service Class did not achieve execution velocity goal

Discussion: When CPExpert detects that a service class did not achieve its performance goal, CPExpert analyzes the basic causes (see the discussion in the above predecessor rules). One of the possible causes of delay is that the service class was delayed because of I/O activity.

- For service classes that are assigned address spaces (that is, the service classes are not transactions managed by a work manager), the SRM does not collect I/O delay information[1]. Rather, any I/O delay is reflected in the UNKNOWN category of delay.

  For these service classes, CPExpert must estimate the I/O delay based on information from SMF Type 72 records and SMF Type 74 records (and potentially, SMF Type 30 records). This rule (Rule WLM350) describes these situations.

- For service classes that describe transactions managed by a work manager (possible with CICS/ESA Version 4.1 and IMS/CICS Version 5), the work manager provides the Workload Manager with information about

[1] Recall that this finding applies for data prior to OS/390 Version 1 Release 3.

I/O delays from the perspective of the work manager. This situation is described in Rule WLM124.

When the UNKNOWN delay is greater than the WLMSIG guidance variable in USOURCE(WLMGUIDE), CPExpert analyzes several possible causes of delay outside the control of the SRM. Initially, CPExpert examines the I/O counts contained in the SMF Type 72 records for the measurement interval. CPExpert divides the I/O service units (R723CIOC) by the I/O service coefficient (SMF72ISD). This yields the total number of I/O operations for the service class during the measurement interval.

CPExpert cannot tell from the Type 72 information whether the I/O operations were directed to tape, to DASD, or to other device types. However, DASD normally is the fastest medium. If the I/O had been directed to DASD, the delay normally would be less than if the I/O had been directed to other activity. CPExpert makes an assumption that all I/O activity had been directed to DASD, simply to get a "feel" as to whether the I/O activity could be a significant cause for delay.

If the DASD Component of CPExpert is not licensed, CPExpert uses the average I/O response time in the measurement interval, for all DASD devices in the configuration.

- CPExpert processes the Type 74 records for DASD devices and computes the overall average device characteristics (I/O response, disconnect time, connect time, PEND time, and I/O Supervisor queue time) for all DASD devices.

- The overall average DASD I/O response time is multiplied by the number of I/O operations generated by the service class missing its performance goal. The result is an estimate of the maximum I/O delay using the overall average DASD response time.

If the DASD Component of CPExpert is licensed and if the CPExpert modification has been made to MXG or MICS to collect Type 30(DD) information for service classes, CPExpert can focus on specific DASD devices used by the service class missing its performance goal.

- CPExpert processes the DASD30DD records created by the modification to MXG or MICS, extracting DASD device information for the service class missing its performance goal.

- CPExpert then processes the Type 74 records for the DASD devices referenced by the service class. CPExpert extracts the device characteristics (I/O response, disconnect time, connect time, PEND time, and I/O Supervisor queue time) for each DASD device referenced by the service class.

- The DASD I/O response time for each device referenced by the service class is multiplied by the number of I/O operations directed to the DASD device to yield an estimate of the I/O delay for each device. CPExpert sums the estimated delays for each device referenced by the service class to yield an overall estimated maximum DASD delay.

CPExpert produces Rule WLM350 if the estimated maximum DASD delay is greater than the actual response time multiplied by the WLMSIG guidance variable. CPExpert provides information showing the average I/O operations per transaction (from the Type 72 records for the service class), the estimated total maximum DASD delay time, and the DASD I/O characteristics during the measurement interval (I/O response, disconnect time, connect time, PEND time, and I/O Supervisor queue time).

There are several considerations with this analysis approach:

- The I/O operations counted in the Type 72 records may not have been directed to DASD. If the I/O operations were directed to some other medium (e.g., they were directed to tape), the analysis might significantly underestimate the effect of I/O on performance. This is because tape I/O operations often are much longer than DASD I/O operations. Consequently, CPExpert might not produce Rule WLM350 if the estimated DASD I/O time were less than the significance factor. Unfortunately, there is no information at present that describes tape I/O delays.

- The DASD I/O operations might be buffered or overlapped with each other. This is quite likely to be the case if the service class handles batch jobs, for example. This situation is less likely with TSO interactive transactions, as TSO interactive transactions often execute few I/O operations and these are often unbuffered and unoverlapped.

- If the overall average DASD I/O response time is used, it may be that the service class referenced DASD devices that experienced I/O response times significantly different from the overall average. This situation would not occur if the CPExpert modification had been made to MXG or MICS, as CPExpert would use DASD I/O information only for the devices referenced by the service class.

- Even if the CPExpert modification has been made to MXG or MICS to collect DASD I/O activity by device, the Type 30 I/O activity counts may not relate well to actual DASD activity due to inconsistencies in how the Type 30 I/O counts are provided to SMF by subsystems.

As a result of the above considerations, the results of the DASD I/O analysis must be viewed with some caution. However, analysts are mostly interested in finding significant delays.

- If Rule WLM350 shows that the estimated I/O delays are very significant, it is quite likely that I/O delays are indeed accounting for much of the UNKNOWN delay. The I/O delay may not be caused by DASD but could be caused by some other (slower) medium.

- If Rule WLM350 is not be produced for a service class with a response goal, you can be reasonably confident that DASD I/O operations are not significantly delaying the service class. If tape (or other relatively slow medium) is causing I/O delays, the service class likely describes batch jobs or long-running started tasks. These service classes do not normally have response goals and thus would not be analyzed in Rule WLM350 code.

- The "bottom line" is that when Rule WLM350 is produced, it is pretty likely that DASD I/O is significantly delaying the response time of the associated service class. The actual data reported may be suspect, but the overall finding likely is correct.

The following example illustrates the output from Rule WLM350:

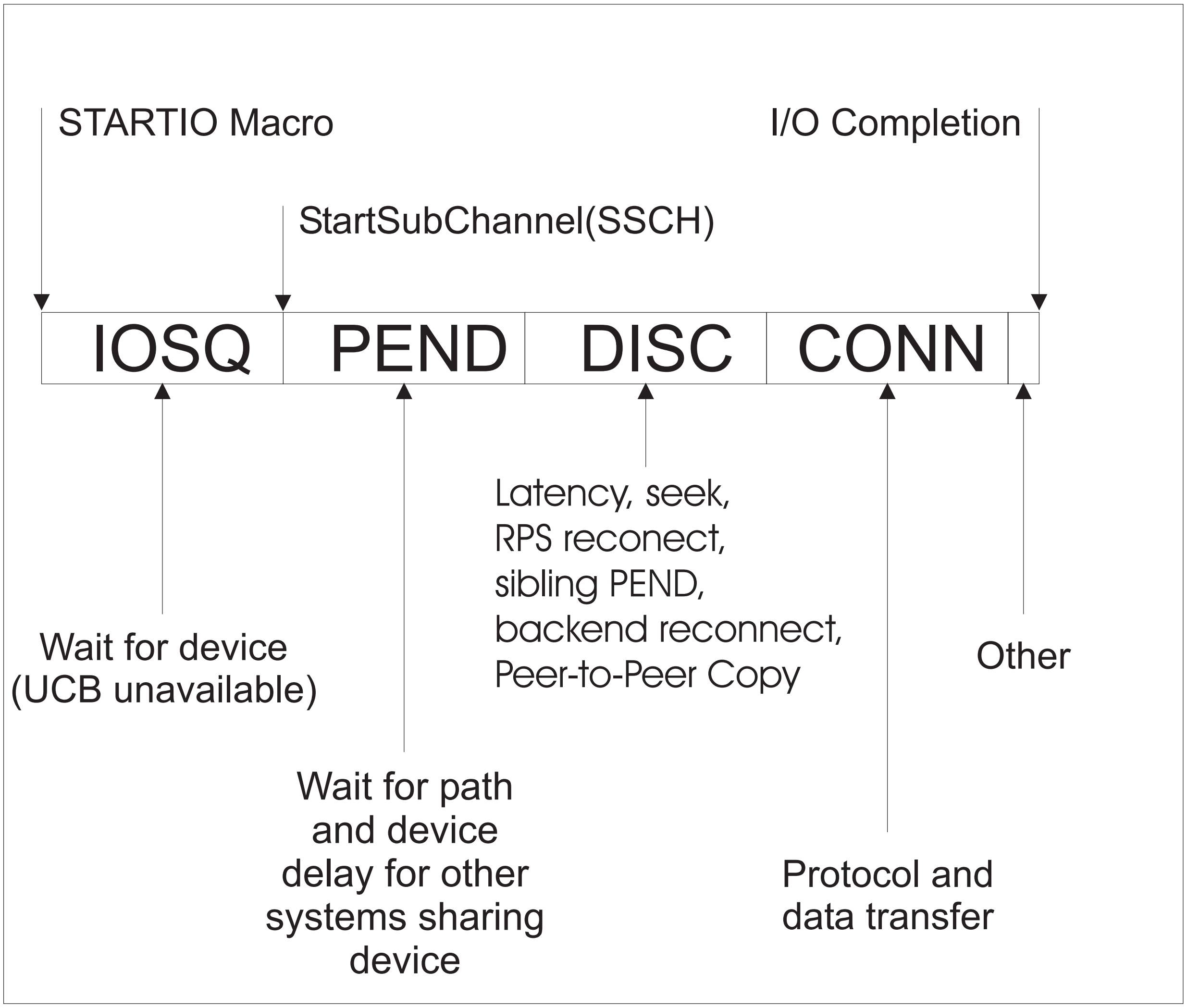
```
RULE WLM350: I/O ACTIVITY MAY HAVE CAUSED SIGNIFICANT DELAYS

   A significant part of the UNKNOWN delay probably can be attributed to
   I/O delay.  CPExpert used the average DASD I/O response time during the
   times when Service Class TSO (Period 1) missed its service goal. The
   average DASD I/O response time was multiplied by the average number of
   I/O operations per transaction to estimate the potential delay that
   might be caused by I/O activity.  The below data shows intervals when
   DASD I/O delay could have caused TSO to miss its service goal:

                            AVERAGE   ESTIMATED
                           I/O COUNT TOTAL DASD  ---AVERAGE DASD I/O TIMES--
   MEASUREMENT INTERVAL   PER TRANS TIME/TRANS  RESP  DISC  CONN  PEND  IOSQ
   13:02-13:07,21JUN1994       5       0.028   0.006 0.003 0.002 0.000 0.000
   13:07-13:12,21JUN1994       5       0.044   0.009 0.004 0.003 0.000 0.002
   13:17-13:22,21JUN1994       5       0.045   0.010 0.004 0.005 0.000 0.000
   13:22-13:27,21JUN1994       5       0.047   0.009 0.004 0.005 0.000 0.000
```

Suggestion: From a high-level view, there are four key measures of DASD performance: IOS Queue (IOSQ) time, pending (PEND) time, disconnect (DISC) time, and connect (CONN) time. These measures are reported by RMF in SMF Type 74 records.

The following figure illustrates these four measures and another potential element of DASD I/O time, titled "Other".



- **IOSQ time**. IOSQ time is the time from the issuance of a STARTIO macro until the StartSubChannel (SSCH) instruction is issued. After the STARTIO macro is issued, the software determines whether the device is busy with *this system*; that is, whether there is an available Unit Control Block (UCB) for the device. If the device is not busy with *this system* (a UCB is available), the SSCH instruction is issued. However, if the device is busy with *this system*, the I/O request is queued. Thus, IOSQ time always means that the device is unable to handle additional requests from *this system*. (The emphasis on "this system" is explained in the below discussion of PEND time.)

  This discussion of IOSQ time does not always apply to Parallel Access Volumes (PAVs)[2]. With PAV devices, MVS creates multiple UCBs for

[2] PAV devices are available with Enterprise System Storage (ESS). With PAV devices, a "base device" address is defined, and a UCB is associated with this base address. "Alias device" addresses can be defined and UCBs are associated with the alias device addresses.

each device, depending on how many “alias devices” have been defined. The multiple UCBs allow multiple active concurrent I/Os on a given device when the I/O requests originate from the same system[3]. Using PAVs can dramatically improve I/O performance by nearly eliminating IOSQ.

Beginning with OS/390 Version 2 Release 4, IOSQ time for service class periods is available in SMF Type 72 records as field R723CIOT.

- **PEND time**. PEND time is the time from the issuance of the StartSubChannel (SSCH) instruction until the device is selected by the control unit and physical positioning commands (such as seek and set sector) are transferred to the device. With modern fixed block architecture (FBA) devices, the PEND time ends when the physical positioning commands are presented to the *logical volume control block* within the control unit. The PEND time is caused by queuing for the path (wait for channel, wait for director port, wait for control unit, wait for device, or wait for “other” reasons)[4].

  The PEND time can be caused by the device being busy from *another system*. In this case, the system issuing the STARTIO macro (*this system*) would have no knowledge that the device was busy with another system. Rather, if a UCB were available for the device, the SSCH would be issued. However, the device could not necessarily be selected (unless multiple allegiance were available), since the device would be busy from another system. Additionally, PEND time could accumulate even with PAV devices if the access were to an extent that was busy with another I/O operation from *this system*.

  PEND time for service class periods is available in SMF Type 72 records (field R723CIWT[5]).

- **DISC time**. DISC means that there is some delay that is often (but not always) associated with a mechanical movement during which the device disconnects from the control unit.

[3] Multiple Allegiance allows multiple active concurrent I/O operations on a given device when the I/O requests originate from different systems.

[4] PEND time is significantly reduced with FICON channels. FICON channels can have multiple I/O operations concurrently active, which reduces the potential PEND time caused by channel busy. There is no port busy time with FICON switches, and control unit time is significantly reduced. This statement regarding PEND time is not necessarily correct if a large number (more than 5) I/O operations are concurrently executing on a FICON channel. Dr. H. Pat Artis and Mr. Robert Ross have presented the results of research indicating that performance degrades significantly when more than 5 I/O operations are concurrently active on a FICON channel (see “Understanding FICON Channel Path Metrics”at www.perfassoc.com).

[5] While the SMF documentation described R723CIWT as “queue time + pending time, the “queue time” refers to queuing for controller, rather than IOSQ. This meaning has been confirmed by IBM SRM/WLM developers and by RMF developers. IOSQ time was added in OS/390 Version 2 Release 4 by the SMF Type 72 field R723CIOT.

With legacy systems (e.g., 3380 drives attached to 3990-2 control units), the DISC time of most concern was associated with seek (arm movement) and rotational position sensing (time waiting for the disk platter to rotate to the location where desired data resides). Considerable performance improvement efforts were directed at reducing the seek activity and reducing the rotational position sensing (RPS)[6] delays for the legacy systems. These two mechanical delays still exist for most modern *redundant array of independent disks* (RAID)[7] systems, but their impact can not be directly reduced with normal methods.

With modern disks, data is cached into Actuator Level Buffers (ALBs), that contain data read from a track on the disk platter. Using ALBs eliminated the RPS delays, since required data is read into the device buffer during a single rotation and stored until a path is available to transfer the data.

Additionally, data is cached into increasingly large cache on the controller. For a read operation, desired data often is found in the cache. Write operations normally end as the date to be written is placed in the cache; and the storage processor writes the data to the device asynchronous with other activity (as a “back end” staging operation).

Consequently, DISC time for modern systems is a result of *cache read miss* operations, potentially back-end staging delay for write operations, peer-to-peer remote copy (PPRC) operations, and other miscellaneous reasons[8]. DISC time often can be very small with adequate cache. For example, there would be zero disconnect time for a cache read hit (the record was found in the cache).

DISC time for service class periods is available in SMF Type 72 records (field R723CIDT).

- **CONN time**. CONN time includes the data transfer time, but also includes protocol exchange[9] (or "hand shaking") between the various components at several stages of the I/O operation.

---

[6] RPS delays were caused by a path not being available when the required data came under a device read head. Since a path was not available, the data could not be read and another rotation of the platter was experienced until the data again came under the device read head. Multiple rotations might be required, depending on the busy level of the path.

[7] An array is an ordered collection of physical devices (disk drive modules) that are used to define logical volumes or devices.

[8] Artis has described a “sibling PEND” condition that results from collisions within the physical disk subsystem of RAID devices. See “Sibling PEND: Like a Wheel within a Wheel,” www.cmg.org/cmgpap/int449.pdf.

[9] Note that the protocol exchange occurs at multiple points in the normal I/O operation, even though it is shown only once in this exhibit.

For devices attached to paths that include parallel channels and ECON channels, the data transfer time is simply the number of bytes transferred divided by the transfer speed. This is because a parallel channel or ESCON channel can have only one data transfer operation in execution at one time.

For devices attached to paths that include FICON channels, the algorithm is more complicated. This primarily is because a FICON channel can perform multiple data transfer (read and write) operations at one time. The data packets for multiple read or write operations are interleaved (or multiplexed) in the FICON link. CONN time for an individual I/O begins with the first frame of data transferred and ends last frame of data transfer, even though data for other I/O operations might be transferred concurrently on the link. Consequently, if multiple data packets (representing data for multiple read or write operations) are interleaved on the FICON link, the elapsed time for any particular I/O operation can be elongated[10] when compared with the elapsed time of the same I/O operation on an ESCON channel.

CONN time for service class periods is available in SMF Type 72 records (field R723CICT).

- **OTHER time**. There are at least two other potential I/O delays for DASD: (1) waiting for the I/O completion interrupt to be serviced by a processor and (2) waiting for the I/O interrupt to be serviced by a domain under PR/SM. Neither potential I/O delay is expected to be of the magnitude of the four "standard" I/O delays. However, they can be significant in special circumstances.

  - Multi-processor configurations can use any processor to service an I/O interrupt. However, when a processor services an I/O interrupt, the processor's high-speed cache storage is no longer valid when control is returned to the interrupted task. Consequently, many of the processor's high-performance design features may be nullified.

    A hardware feature allows processors to be disabled for I/O interrupts. With this method, only a small number (perhaps only one) processor is enabled for interrupt processing. Only this processor will have its high-speed cache storage disturbed by the task-switching required for interrupt processing, and only this processor will periodically have its high-performance design features nullified. The disadvantage to this

[10] The relative speed of a FICON channel is much higher than that of an ESCON channel. Consequently, the elapsed time of any particular I/O operation should be less on a FICON channel than on an ESCON channel, even if there are multiple I/O operations interleaving data. This statement regarding elapsed time is not necessarily correct if a large number (more than 5) I/O operations are concurrently executing on a FICON channel. Dr. H. Pat Artis and Mr. Robert Ross have presented the results of research indicating that performance degrades significantly when more than 5 I/O operations are concurrently active on a FICON channel (see “Understanding FICON Channel Path Metrics”at www.perfassoc.com).

approach is that an interrupt may occur while the processor is busy servicing a previous interrupt.

If an interrupt is pending and no processor is enabled to service the interrupt, the interrupt must wait until a processor is available. This time should be insignificant, unless the system is processing a significantly large number of I/O operations. If the system is processing a large number of I/O operations (or if the I/O is particularly time-sensitive), the interrupt pending delay could pose performance problems.

After the processor completes processing for an I/O interrupt, it issues a Test Pending Interrupt (TPI) instruction to determine whether there are any interrupts pending. If an I/O interrupt is pending, the processor proceeds to service that interrupt.

The CPENABLE keyword in the IEAOPTxx member of SYS1.PARMLIB is used to specify the percent of I/O interrupts detected by the TPI instruction, compared with all I/O interrupts. When the percent exceeds the high threshold of the CPENABLE keyword, MVS enables another processor to handle pending I/O interrupts. If the percent falls below the low threshold of the CPENABLE keyword, MVS will disable a processor (to the point that only one processor is enabled). IBM's recommended setting for the CPENABLE keyword differs, depending on the level of processor.

- MVS environments running under as a guest under VM or in a logical partition (LPAR) under PR/SM are subject to I/O interrupt delays. These delays can occur if another guest (for VM) or another domain is in its dispatch interval when the I/O interrupt completion is posted. The I/O interrupt remains pending until the guest or domain is dispatched. These delays have been estimated to be far more significant than might otherwise be expected.

OTHER time for service class periods is not available in SMF Type 72 records.